



TITLE:

# 組合せ構造の列挙とサンプリング (代数、形式言語、計算システム理 論とその応用)

AUTHOR(S):

松井, 泰子

---

CITATION:

松井, 泰子. 組合せ構造の列挙とサンプリング(代数、形式言語、計算システム理論とその応用). 数理解析研究所講究録 2007, 1562: 92-98

ISSUE DATE:

2007-06

URL:

<http://hdl.handle.net/2433/81101>

RIGHT:

# 組合せ構造の列挙とサンプリング

東海大学理学部情報数理学科 松井 泰子 (Yasuko Matsui)  
Department of Mathematical Sciences  
Tokai University

## 1 列挙とサンプリング

与えられた集合  $\mathcal{F}$  の要素で、性質  $P$  を満たすものを全て求めることを列挙 (enumeration) といい、列挙を行う問題を列挙問題 (enumeration problem) という。列挙問題においては、解を 1 つ見つけることは一般に容易である (解を 1 つ求めるのも  $NP$  困難である列挙問題も存在する)。しかし、解を全て見つけたという保障をすることは容易では無い。また、単純にメモリに解を蓄え、解が 1 つ見つかる度に付き合わせて重複しているか否かをチェック等すると、解集合の大きさは指数個にのぼり、すぐにメモリ限界に達する。そのため、高速に解を列挙しかつメモリを効率良く使用するアルゴリズムの開発が望まれるようになった。列挙問題を解くアルゴリズムは列挙アルゴリズムと呼ばれ、1960 年代頃から研究されている。当初は理論的な興味から列挙アルゴリズムが提案されたが、近年、コンピュータの処理能力の飛躍的な進歩とメモリの低廉化により、列挙アルゴリズムの実装が可能となり、列挙問題に対する列挙アルゴリズムの構築が注目を集めている。

列挙アルゴリズムの計算時間の算定は、通常、問題の入力サイズと解の出力数  $N$  の関数によるオーダーで表される。計算時間が問題の入力サイズと  $N$  の多項式のオーダーでおさえられる時、列挙アルゴリズムは出力多項式時間 (output polynomial) であると呼ばれる。特に、解を 1 つ出力してから次の解の出力に要する時間が、問題の入力サイズの多項式のオーダーでおさえられる時、アルゴリズムは多項式時間遅延 (polynomial delay) であるという [18]。列挙アルゴリズムの効率化に関しては、計算時間の短縮以外に、出力の大きさを縮小するコンパクト出力 (compact output) の研究も行われている。解を陽に出力するのでは無く、直前の解との差分のみを出力するなどの効率化を図るのである。

列挙アルゴリズムで用いられる代表的な枠組みは、バックトラック法、分割探索法、逆探索法 (reverse search) [2] 等が挙げられる。ここでは本稿で扱う、逆探索法について触れる。逆探索法は、各頂点が列挙される解に対応する木 (列挙木) を陰に構築し、列挙木の上を探索して解を列挙する方法である。列挙木は、頂点間の親子関係によって定義される。すなわち、列挙木の根に対応する頂点以外では、自分の親となる頂点が一意に定義できるといえるものである。一般にそのような親子関係は容易には定義できない。列挙木の構築は、親が子の候補 (子であるものを全て含む) が全て探索でき、子からは自分の親が同定できれば実現できる。逆探索法では、列挙木の根に対応する頂点から順に子の候補を探索し、子の候補が見つかる度に、子から親を同定して親子関係をチェックし、親ならば子 (解) を出力するという操作を、子の候補が探索される限り再帰的に繰り返す。この繰り返しにより、列挙木を陽に構築することなく全ての頂点が探索できるのである。逆探索法に関しては、[12] に提案者による日本語の解説がある。列挙アルゴリズムの詳細に関しては、文献 [20] を参照されたい。

列挙アルゴリズムが提案される一方、性質  $P$  を満たすものをランダムにサンプリングしたいという要求がある。列挙されるものの個数は指数個に膨れ上がるため、そのうちの一部を求めて利用するというものである。例えば、統計で分割表のデータを検定する時に、周辺和を固定した分割表をサンプリングして推定量を計算する。分割表のデータを正確に検定する場合は、周辺和を固定した分割表を全て列挙する必要があるが、その総数が膨大となるため、列挙の代わりに一様ランダムにサンプリングした分割表を用いて検定を行う。そのため、周辺和を固定した分割表のサンプリングを、マルコフ連鎖モンテカルロ法 (MCMC 法) 等によって実現するのである。近年は計算量の観点から、マルコフ連鎖の攪拌時間 (mixing time) が問題の入力サイズ、誤差、分割表の総数による多項式のオーダーでおさえられる、実用的なマルコフ連鎖の構築等の研究が盛んである [1],[3],[4],[5], [7],[9],[10], [14],[15],[17], [19],[23],[27].

## 2 コーダルグラフの perfect sequence の列挙

本稿では、以降においてコーダルグラフ上で定義できる perfect sequence に対する列挙アルゴリズムを提案する。コーダルグラフは幅広い応用が知られるグラフである。例えば、統計ではコーダルグラフは分解可能モデルや、ベイジアンネットワークにおける周辺確率の局所計算等にも活用されている。数理工学においては、半正定値計画問題、信号処理、数値計算、準ニュートン法等でコーダルグラフが重要な役割を果たしている。また、一般のグラフ上では解決が困難な問題が、コーダルグラフ上では比較的容易に解ける場合もある。このような背景から、コーダルグラフに関する研究が、多様な分野の研究者の注目を集めている。ここでは、コーダルグラフに付随した perfect sequence という順列について考える。与えられたコーダルグラフの perfect sequence の列挙によって、統計や可換代数の諸問題が解決できる [14] [16]. そのため perfect sequence の列挙に対する要求が高いが、列挙される個数が指数オーダーとなり、列挙されたものをメモリに蓄える単純な方法では列挙は容易でない。既存の研究では perfect sequence をランダム生成するアルゴリズムは存在するが、列挙アルゴリズムは (筆者の知識の限りでは) 未だ提案されていない。

以上のような背景から、本稿では perfect sequence を列挙するアルゴリズムを提案する。まず諸定義を行う。  $V$  を頂点集合、  $E$  を辺集合とするグラフ  $G = (V, E)$  において、  $G$  中の長さ 4 以上の閉路が必ず弦をもつときコーダルグラフとよぶ。  $\mathcal{C}, \mathcal{S}$  を各々  $G$  中の極大クリーク (maximal clique) の集合と極小頂点分離 (minimal vertex separator) 集合とする。木  $T = (\mathcal{C}, \mathcal{E})$  が  $G = (V, E)$  のクリーク木であるとは、任意の二つの極大クリーク  $C_1, C_2 \in \mathcal{C}$  と、  $T$  上で  $C_1, C_2$  を結ぶ一意なパス中にある極大クリーク  $C_3 \in \mathcal{C}$  に対し、  $C_1 \cap C_2 \subseteq C_3$  が成り立つことである。  $G$  がコーダルグラフである必要十分条件は、  $G$  にクリーク木が存在することである [13]. 一般に、コーダルグラフにはクリーク木が複数存在する。Kumar and Madhavan は、クリーク木の総数を数え上げるアルゴリズムを提案している [21]. クリーク木  $T = (\mathcal{C}, \mathcal{E})$  上で隣接する任意の二つの極大クリーク  $C_1, C_2 \in \mathcal{C}$  について、  $C_1 \cap C_2 = S$  となる極小頂点分離集合  $S \in \mathcal{S}$  が存在する。  $G$  中の極大クリークを  $C_k, i = 1, \dots, K$  とし、添え字集合を  $\mathcal{I} = \{1, \dots, K\}$  とする。置換  $\pi: \mathcal{I} \rightarrow \mathcal{I}$  について、  $H_{\pi(k)}, k = 1, \dots, K$  と  $S_{\pi(k)}, k = 2, \dots, K$  を各々  $H_{\pi(k)} = C_{\pi(1)} \cup \dots \cup C_{\pi(k)}, S_{\pi(k)} = H_{\pi(k-1)} \cap C_{\pi(k)}$  と定義す

る。列  $\pi: C_{\pi(1)}, C_{\pi(2)}, \dots, C_{\pi(K)}$  が極大クリークの perfect sequence であるとは、各  $S_{\pi(k)}$  がクリークであり、すべての  $k \geq 2$  について  $S_{\pi(k)} \subset C_{\pi(k')}$  となる  $k' < k$  が存在することである。極大クリークの perfect sequence が存在する必要十分条件は、 $G$  がコードルグラフであり、すべての  $k$  について  $S_{\pi(k)} \in \mathcal{S}$  で、 $\{S_{\pi(2)}, \dots, S_{\pi(K)}\} \in \mathcal{S}$  を満たすことである。ただし、 $S_{\pi(2)}, \dots, S_{\pi(K)}$  には同じ極小頂点分離集合  $S$  が複数現れるかもしれない。

2006 年、原と竹村は、与えられたコードルグラフから perfect sequence をランダム生成するためのマルコフ連鎖の構築法を提案した [14]。彼らが提案したマルコフ連鎖は連結な 2 部グラフ  $B = (V_1, V_2, E)$  である。頂点集合  $V_1, V_2$  は各々、クリーク木の集合と perfect sequence の集合に対応している。クリーク木  $u \in V_1$  と perfect sequence  $v \in V_2$  が相互に構築可能である場合、有向辺  $(u, v), (v, u) \in V_1 \times V_2$  が存在するグラフである。一般に 1 つのクリーク木から構成できる perfect sequence は複数存在する。また 1 つの perfect sequence から構成できるクリーク木も複数存在し、2 部グラフ  $B$  は連結である。彼らのアイデアは、前出のように構築したマルコフ連鎖上を、 $V_1$  から  $V_2$  へ、 $V_2$  から  $V_1$  へと交互にランダムに遷移して perfect sequence をサンプリングすることである。遷移は、Lauritzen によって提案された 2 つのアルゴリズム [22]、すなわち、(1) 任意のクリーク木から perfect sequence  $\pi$  をランダム生成するアルゴリズム、(2) 任意の perfect sequence  $\pi$  からクリーク木をランダム生成するアルゴリズム、を適用している。

以降では、与えられたコードルグラフから perfect sequence を列挙する、以下のような列挙問題を考える。

**Problem AllPerfectSequences**  
 INPUT: コードルグラフ  $G = (V, E)$   
 OUTPUT:  $G$  の perfect sequence  $\pi$  を重複無く全て。

上の列挙問題に対し、次のような列挙を考えるのが自然であろう。まず、コードルグラフ  $G$  のクリーク木を列挙し、クリーク木が得られるたびにその perfect sequence を列挙するのである。しかし、この方法では perfect sequence の重複は避けられない。従って、列挙アルゴリズムの枠組みの適用が望まれる。提案する列挙アルゴリズムでは、perfect sequence の重複を避けるために「逆探索法」の枠組みを用いて実現する。

## 2.1 コードルグラフのクリーク木の列挙

まず、コードルグラフからクリーク木を列挙する列挙問題を考える。

**Problem AllCliqueTrees**  
 INPUT: コードルグラフ  $G = (V, E)$   
 OUTPUT:  $G$  のクリーク木  $T = (C, \mathcal{E})$  を重複無く全て。

コードルグラフから生成される、クリークグラフと呼ばれる重み付きグラフの概念を導入すると、列挙問題 AllCliqueTrees は、以下に挙げるクリークグラフ中の最大全張木の列挙問題 AllMaximumSpanningTrees に帰着できる。コードルグラフ  $G = (V, E)$  のクリー

クグラフ  $G_c = (\mathcal{C}, \mathcal{E}_c)$  とは,  $G_c$  中の各頂点  $v \in \mathcal{C}$  は  $G$  中の極大クリークに一対一対応し,  $G$  中の極大クリーク  $C_1, C_2 \subseteq \mathcal{C}$  について  $C_1 \cap C_2 \neq \emptyset$  が成り立つとき,  $C_1, C_2$  に対応する  $G_c$  の頂点  $v_1, v_2 \in \mathcal{C}$  間を辺  $\{v_1, v_2\} \in \mathcal{E}_c$  で結んだグラフである. 辺  $\{v_1, v_2\} \in \mathcal{E}_c$  の重みは,  $|C_1 \cap C_2|$ , すなわち  $C_1, C_2$  の極小分離集合の頂点数で与える. 最適化の分野では, コーダルグラフ  $G$  のクリークグラフ  $G_c$  の最大全張木は, コーダルグラフ  $G$  のクリーク木  $T$  に一対一対応することが知られている. よって, 列挙問題 **All\_CliqueTrees** は, 以下のコーダルグラフ  $G$  のクリークグラフ  $G_c$  中の最大全張木の列挙問題 **All\_MaximumSpanningTrees** と等価である [21].

**Problem All\_MaximumSpanningTrees**

INPUT: コーダルグラフ  $G = (V, E)$  のクリークグラフ  $G_c = (\mathcal{C}, \mathcal{E}_c)$

OUTPUT:  $G_c$  の最大全張木を重複無く全て.

列挙問題 **All\_MaximumSpanningTrees** は, 重み付きグラフ中の最大全張木の列挙アルゴリズム ([11] 等) を適用することで実現出来る. 最大全張木の列挙に要する総計算時間は  $O(|\mathcal{C}| + |\mathcal{E}_c| + N\sqrt{|\mathcal{E}_c|})$  である. ただし,  $N$  は列挙する最大全張木の総数である.

## 2.2 クリーク木の perfect sequence の列挙

ここでは, 列挙問題 **All\_MaximumSpanningTrees** で得られる最大全張木に対応するクリーク木から, perfect sequence  $\pi$  を列挙するために, 以下の列挙問題を考える.

**Problem All\_PerfectSequences\_from\_CliqueTree**

INPUT: コーダルグラフ  $G = (V, E)$  のクリーク木  $T = (\mathcal{C}, \mathcal{E})$

OUTPUT:  $T$  の perfect sequence  $\pi$  を重複無く全て.

クリーク木  $T$  の perfect sequence  $\pi$  は, 以下の操作で構成できる.  $T$  中の任意の頂点を根とし, 根から葉に向けて辺に向き付けをした根付き木  $T'$  を作ると,  $T'$  上でトポロジカル順序  $\sigma$  が定義できる. このトポロジカル順序  $\sigma$  は perfect sequence  $\pi$  の定義を満たす. よって, 列挙問題 **All\_PerfectSequences\_from\_CliqueTree** は以下の列挙問題 **All\_TopologicalOrdering** と等価である.

**Problem All\_TopologicalOrdering**

INPUT: コーダルグラフ  $G = (V, E)$  のクリーク木  $T = (\mathcal{C}, \mathcal{E})$

OUTPUT:  $T$  から構成できる根付き木上のトポロジカル順序を重複無く全て.

列挙問題 **All\_TopologicalOrdering** は, 非巡回的グラフのトポロジカル順序を列挙する列挙アルゴリズム ([2],[26]) で実現できる. トポロジカル順序の列挙に要する総計算時間は  $O(|\mathcal{C}| + N'|\mathcal{C}|^2)$  である. ただし,  $N'$  は列挙するトポロジカル順序の総数である. 前出の2つの列挙問題 **All\_MaximumSpanningTrees**, **All\_MaximumSpanningTrees** を解くことで, perfect sequence を全て求めること可能となるが, 明らかに  $\pi$  は重複して出力されるため, それを避ける枠組みの提案が必要となる. 本稿では, 逆探索法で用いられる

解の間の「親子関係」を利用することで、重複を避けた列挙を実現する。クリーク木を親, perfect sequence を子とする親子関係である。一般に1つの perfect sequence から複数のクリーク木が生成できるが, それらの中から一意な親を同定する。具体的には次の操作を行う。クリーク木  $T$  において,  $T$  中の頂点を1つ根と定めて根付き木  $T'$  を作り,  $T'$  上のトポロジカル順序  $\sigma$  を列挙する。トポロジカル順序  $\sigma$  が得られる度に, 以下に説明するようにして  $\sigma$  から一意なクリーク木  $T^*$  を作成する。  $T^* = T$  ならば,  $\sigma$  を  $T$  の perfect sequence として出力する。まず, クリークグラフ  $G_c$  の頂点をトポロジカル順序  $\sigma(C_{\pi(1)}, C_{\pi(2)}, \dots, C_{\pi(K)})$  に従って並べ, 各クリークグラフの辺  $\{C_{\pi(i)}, C_{\pi(j)}\}$  に向き付けをして有向辺  $(C_{\pi(i)}, C_{\pi(j)})(i < j)$  とし, 辺の重みは  $\mathcal{E}_c$  と同じである重み付き有向グラフ  $G'$  を作成する。次に,  $G'$  の根以外の各頂点において, 頂点に入る有向辺から辺上の重みが最大であるものを選ぶ。そのような辺が複数存在する場合は, そのうち添え字の1番小さいものを選ぶことにする。以上のようにして辺を選ぶと, クリークグラフから木  $T'$  が一意に得られ, 木  $T'$  の辺を無向にしたグラフ  $T^*$  はクリーク木である。

最後に, 提案した列挙アルゴリズムの計算量と記憶容量を示す。

**定理** [松井 [25]] 与えられたコーダルグラフ  $G$  の perfect sequence の列挙は,  $G$  のクリーク木  $T$  と perfect sequence の間に親子関係を導入することで実現できる。列挙に要する計算時間は  $O(NN^*(|C|^2 + |\mathcal{E}_c|))$ , 記憶容量は  $O(|C||\mathcal{E}_c|)$  である。ただし,  $C$  は  $G$  のクリークグラフの頂点数,  $|\mathcal{E}_c|$  は  $G$  のクリークグラフの辺数,  $N$  は列挙されるクリーク木の総数,  $N^*$  は列挙される perfect sequence の総数である。■

## 参考文献

- [1] D. ALDOUS, Random walks on finite groups and rapidly mixing Markov chains, in A. Dold and B. Eckmann, eds., *Séminaire de Probabilités XVII 1981/1982*, vol. 986 of Springer-Verlag Lecture Notes in Mathematics, Springer-Verlag, New York, (1983), pp. 243–297.
- [2] D. AVIS AND K. FUKUDA, Reverse search for enumeration, *Discrete Appl. Math.*, 65(1996), pp.21-46.
- [3] R. BUBLEY AND M. DYER, Path coupling: A technique for proving rapid mixing in Markov chains, *38th Annual Symposium on Foundations of Computer Science*, IEEE, San Alimitos, (1997), pp. 223–231.
- [4] R. BUBLEY *Randomized Algorithms : Approximation, Generation, and Counting*, Springer-Verlag, New York, 2001.
- [5] F. R. K. CHUNG, R. L. GRAHAM, AND S. T. YAU, On sampling with Markov chains, *Random Structures and Algorithms*, 9 (1996), pp. 55–77.

- [6] P. DIACONIS AND A. GANGOLLI, Rectangular arrays with fixed margins, in D. ALDOUS, P. P. VARAIYA, J. SPENCER, AND J. M. STEELE (Eds.), *IMA Volumes on Mathematics and its Applications*, 72 (1995), pp. 15–41, Springer, New York.
- [7] P. DIACONIS AND L. SALOFF-COSTE, Random walk on contingency tables with fixed row and column sums, Technical Report, Department of Mathematics, (1995), Harvard University.
- [8] P. DIACONIS AND B. STRUMFELS, Algebraic algorithms for sampling from conditional distributions, *The Annals of Statistics*, 26 (1998), pp. 363–397.
- [9] M. DYER AND C. GREENHILL, Polynomial-time counting and sampling of two-rowed contingency tables, *Theoretical Computer Sciences*, 246 (2000), pp. 265–278.
- [10] M. DYER, R. KANNAN, AND J. MOUNT, Sampling contingency tables, *Random Structures and Algorithms*, 10 (1997), pp. 487–506.
- [11] G. N. FREDERICKSON, Data structures for on-line updating of minimum spanning trees, *SIAM J. Comput.*, 14(1985), pp. 781–798.
- [12] 福田公明, 逆探索とその応用, 離散構造をアルゴリズム, 近代科学社 (1993), pp. 47–78.
- [13] F. GAVRIL, The intersection graphs of subtrees in trees are exactly the chordal graphs, *J. Combin. Theory Ser. B*, 116 (1974), pp. 47–56.
- [14] H. HARA AND A. TAKEMURA, Boundary cliques, clique trees and perfect sequences of maximal cliques of a chordal graph, Technical Report, METR, 2006-41, Department of Mathematical Informatics, University of Tokyo, 2006.
- [15] D. HERNEK, Random generation of  $2 \times n$  contingency tables, *Random Structures and Algorithms*, 13 (1998), pp. 71–79.
- [16] J. HERZOG, T. HIBI, X. ZHENG, Monomial ideals whose powers have a linear resolution, *Math. Scand.*, 95 (2004), pp. 23–32.
- [17] M. R. JERRUM AND A. SINCLAIR, The Markov chain Monte Carlo method: an approach to approximate counting and integration, in *Approximation Algorithm for NP-hard problems*, D. S. HOCHBAUM (Ed.), PWS publishing, Boston, 1997, pp. 482–520.
- [18] D. S. JOHNSON, M. YANNAKAKIS, AND C. H. PAPADIMITRIOU, On generating all maximal independent sets, *Information Processing Letters*, 27(1988), pp. 119–123.
- [19] R. KANNAN, P. TETALI AND S. VEMPALA, Simple Markov chain algorithm for generating bipartite graphs and tournaments, in *8th Annual Symposium on Discrete Algorithms*, ACM-SIAM, San Francisco, California, 1997, pp. 193–200.

- [20] 久保幹雄, 田村明久, 松井知己編, 「応用数理計画ハンドブック」, 朝倉書店 (2002).
- [21] P. S. KUMAR AND C. E. V. MADHAVAN, Clique tree generation and new subclasses of chordal graphs, *Discrete Appl. Math.* 117(2002), pp. 109–131.
- [22] S. L. LAURIZEN, *Graphical models*, Clarendon Press, Oxford (1996).
- [23] T. MATSUI, Y. MATSUI AND Y. ONO, Random generation of  $2 \times 2 \times \cdots \times J$  contingency tables, *Theor. Comput. Sci.*, 326(1-3)(2004), pp. 117–135.
- [24] Y. MATSUI AND T. UNO, On the enumeration of bipartite minimum edge colorings, Graph Theory in Paris, *Proceedings of a Conference in Memory of Claude Berge, Trends in Mathematics*, Birkhaeuser Boston Inc., (2006), pp. 271–285.
- [25] Y. MATSUI, *An algorithm for generating all perfect sequences of a chordal graph*, preprint.
- [26] G. PRUESSE AND F. RUSKEY, Generating linear extension fast, *SIAM Journal of Computing*, 23,2(1994), pp. 373–386.
- [27] A. TAKEMURA AND S. AOKI, Some characterizations of minimal Markov basis for sampling from discrete conditional distributions, *Technical Report METR 2002-04*, Dept. of Mathematical Engineering and Information Physics, Faculty of Engineering, The University of Tokyo, 2002.